Anytime Coevolution of Form and Function

Magdalena D. Bugajska and Alan C. Schultz
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory, Code 5515
Washington, DC 20375
{magda, schultz}@aic.nrl.navy.mil
tel. (202) 404-4946 fax. (202) 767-2166

Abstract- This paper describes an approach to continuous coevolution of form (the morphology) and function (the control behavior) for autonomous vehicles. This study focuses on coevolution of the characteristics such as beam width and range of individual sensors in the sensor suite, and the reactive strategies for collision-free navigation for an autonomous micro air vehicle. The results of the evolution of the system in a fixed simulation model were compared to case-based anytime learning (also called continuous and embedded learning) where the simulation model was updated over time to better match changes in the environment.

1 Introduction

Autonomous vehicles that can change their own morphology on the fly are highly desirable in many domains. For example, the ability of an air vehicle to modify its air frame and the configuration of its control surfaces during certain stages of the flight, such as take offs, attacks, or landings, would have a direct impact on the system's efficiency, performance, and safety. This shape-shifting or morphing mechanism would also be desirable in an Urban Search and Rescue robot to enhance its ability to traverse difficult internal structures within collapsed buildings.

Evolutionary algorithms have been successfully applied to automate the design of robots' morphology, the design of the controllers, and more recently to coevolution of form and function. It is our belief that the natural process of coevolving the form and function of living organisms can be applied to the design of morphology and control behaviors of autonomous vehicles in order to simplify the design process and improve the performance of the system. In our work, coevolution of form and function has been applied to the micro air vehicle (MAV) domain. The design of the sensory payload and the controller for an MAV is complicated by the size of the vehicle (wingspan on the order of 6 inches), its limited payload, and a great variety of possible applications. The design issue addressed explicitly in this study is minimization of power requirements. It is assumed that power efficiency is inversely proportional to the coverage of the sensor suite. The work presented here is an extension of the research published in [Bugajska 2000] and [Bugajska 2002].

In addition, an important problem arising for all autonomous vehicles that are expected to perform tasks for extended periods is how to adapt the components of the system in response to unexpected changes in the environment or in their own capabilities in close to real time. Continuous and embedded learning (also called anytime learning) [Grefenstette 1992] is a general approach to continuous learning in changing environments. The vehicle's learning module continuously tests new strategies in an embedded simulation model which is updated in response to changes in the environment. In the past, this approach has been successfully applied to learning and adaptation of robotic behaviors in dynamic environments as well as in situations where the robot experiences sensor failures. This study focuses specifically on the continuous coevolution of a minimal sensor suite, which allows for most efficient collision-free navigation, in a changing environment. The approaches to evolution in a simulation without feedback from the task environment, are compared to case-based continuous and embedded learning [Ramsey 1994] in a simulation where such feedback exists.

The remainder of this paper briefly outlines the related work and then continues with a description of our implementation of coevolution of the characteristics of a sensor suite and collision-free navigation of an MAV. The simulated environment, aircraft, and sensors are described along with the details of the learning system. Finally, the initial results of the learning experiments in a changing environment are discussed, and the future direction of the research is outlined.

2 Coevolution of Form and Function

In recent years, the result of the evolution of behaviors for autonomous agents in simulation ([Nolfi 1994, Harvey 1992, Schultz 1996, Potter 2001]) and real world ([Floreano 1996]), and research in automation of structural design ([Husbands 1996, Funes 1997, Lichtensteiger 1999, Lund 1997, Mark 1998]), has lead researchers to explore the concept of coevolution of form and function for autonomous agents. [Cliff 1993] and [Cliff 1996] present research on concurrent evolution of neural network controllers and visual sensor morphologies, for visually guided tracking. [Sims 1994] presents a system for the coevolution of morphology and behavior of virtual creatures that compete in a physically simulated three-dimensional world. Similar work is presented in [Hornby 2001] where the body and brain of the creatures are evolved using Lindenmayer systems as generative encoding. In [Lee 1996] a hybrid genetic programming/genetic algorithm approach is presented that allows for evolution of both controllers and robot bodies to achieve behavior-specified tasks. [Balakrishnan 1996] presents the comparative study of evolution of a control system given a fixed sensor suite, and coevolution of sensor

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **Anytime Coevolution of Form and Function**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Navy Center for Applied Research in Artificial Intelligence (NCARAI),4555 Overlook Avenue SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
In Proceedings of 2003 Congress on Evolutionary Computaton (CEC-2003), Canberra, Australia, December 8 - 12, 2003.

14. ABSTRACT
This paper describes an approach to continuous coevolution of form (the morphology) and function (the control behavior) for autonomous vehicles. This study focuses on coevolution of the characteristics such as beam width and range of individual sensors in the sensor suite, and the reactive strategies for collision-free navigation for an autonomous micro air vehicle. The results of the evolution of the system in a fixed simulation model were compared to case-based anytime learning (also called continuous and embedded learning) where the simulation model was updated over time to better match changes in the environment.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

characteristics (placement and range) and the control architecture for the task of box pushing. In previous work [Bugajska 2000] and [Bugajska 2002], we explored coevolution of the beam width of the individual sensors in the sensor suite and the collision-free navigation behavior in context of different controller representations and coevolution approaches in micro air vehicles. This study extends the previous work by exploring the coevolution of form and function in the context of changing environments; we combine the coevolution of form and function with anytime learning technique. In addition, this study extends our previous work by evolving the sensing range of the individual sensors in addition to the beam width.

2.1 Representation

In this study, each individual (chromosome) in the population, contains the genetic material describing the information of both the morphology and the control behavior of the autonomous agent. The characteristics of the sensor suite are encoded in a floating-point vector with elements for beam width and the range of individual sensors in the suite (Section 5.1). The collision-free navigation behavior is represented as a set of stimulus-response rules (Section 4.1).

2.2 Environment

A high-fidelity, 3-D flight simulator (Fig. 1), which includes an accurate parameterized model of a 6-inch MAV, has been used to model the environment and the vehicle. The simulation allows the user to control the aircraft by specifying only the turn rate values; the speed and altitude of the plane are adjusted appropriately by low-level PID controllers. In this study, the MAV is controlled by specifying discreet turning rates between 20 and 20 degrees in 5-degree increments.

The trees (obstacles) are modeled as spheres on top of cylinders. Any contact between the plane and the tree constitutes a collision. The density of trees is user-defined as the number of trees per square foot assuming uniform distribution and varied from 1.25 to 5.0 trees per hundred square feet. At the beginning of each simulated flight, the MAV is placed at a random location within a specified area away from the target. The target is stationary and reachable during every trial.

The simulated MAV has a sensor, which returns the relative range and bearing to the target. It is also equipped with an array of range sensors positioned symmetrically along the direction of flight and radially from the center of the vehicle. Each sensor is capable of detecting obstacles and returning the range to the closest object within its field of view. The beam width and the range of the individual range sensors are evolved along with control behavior.

2.3 Fitness Function

The morphology of the sensor suite and the control behavior of the MAV are evolved in simulation. During each evaluation, a number of episodes is performed that begins with placement of the MAV at a random distance away from the target facing in a random direction, which is followed by a random placement of trees in the environment. The episodes end with either a successful arrival of the MAV at the target location, a loss of the MAV due to energy/time running out, or a loss of the MAV due to collision with an obstacle. The fitness of the individual is based on the quality of the sensor suite and execution of the task and as in our previous work is defined as follows:

Figure 1: The screenshot of the 3-D simulated environment used for the experiments. The white sphere marks the target and dark gray (green) spheres with light gray cylinders mark the obstacles (trees).

if (reached the goal)
  payoff is based on
    the distance MAV traveled (Section 4.3)
    PLUS
    the quality of the sensor suite (Section 5.3)
else if (collision or time out occured)
  payoff based on
    the distance away from target (Section 4.3)

It should be noted that the contribution due to the quality of the sensor suite is considered only once the task performance is satisfactory and that payoff is only assigned at the end of the episode.

3 Continuous and Embedded Learning

The main focus of this study is coevolution of form and function for extended periods in changing environments. Continuous and embedded learning ([Grefenstette 1992, Ramsey 1994]) is a general approach to continuous learning in a changing environment. As shown in Figure 2, the agents learning module continuously tests new strategies against a simulation model of the environment, and dynamically updates the knowledge base (behaviors) used by the agent. The execution module controls the agents interaction with the environment, and includes a monitor process that can dynamically modify the simulation model based on its observation of the environment. When the simulation model is modified, the learning process continues with the modified model. The learning system is assumed to operate indefinitely, and the execution system uses the results

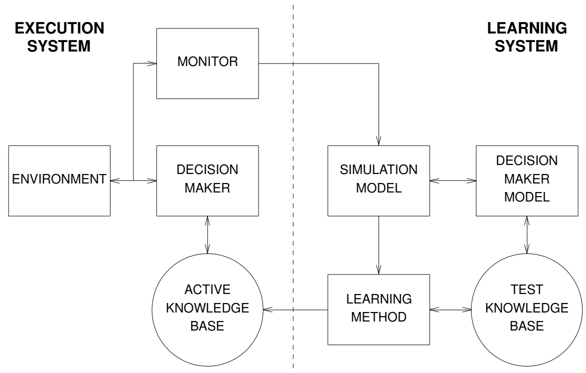


Figure 2: The anytime (also known as continuous and embedded) learning model.

of learning as they become available. This learning approach was previously used to continuously evolve tracking behaviors ([Grefenstette 1992]) and door traversing behavior ([Schultz 2000]) in face of changing environment and changes in the agent's own capabilities such as sensor failures.

In this instantiation of anytime learning, the only measurable aspect of the environment is the density of the obstacles (trees). When the monitor detects the change, the environment model is updated and the learning system is re-initialized. Currently, the system is re-initialized using a combination of previously evolved strategies chosen based on their fitness and the similarity of the model under which they were evolved, and a simple default strategy.

4 Evolution of Function

The performance of the system is determined by the agent's ability to perform the task. In our study, the MAV must be able to efficiently and safely navigate among obstacles (trees) to a target location. The desired behavior should maximize the number of times the MAV reaches the target location while minimizing the distance traveled to that location. Every single evaluation is performed in a randomly created environment (random MAV position and orientation, random, but uniform tree placement, etc.) with complexity determined by the tree density.

4.1 Problem Representation

In this study, the collision-free navigation behavior is implemented as a collection of stimulus-response rules (see [Bugajska 2002] for alternative approach). Each stimulus-response rule consists of conditions that match against the current sensors of the agent, and an action that suggests action to be performed by it. For example, a rule (gene), which states that if there is an obstacle fairly close and roughly ahead of the vehicle, even when the goal is also ahead of it, the vehicle should turn left, could be represented as:

```
RULE 122
IF  sonar4 < 45 AND bearing = [-20, 20]
THEN SET turn_rate = -100
```

Each rule has an associated strength with it as well as a number of other statistics. During each decision cycle, all the rules that match the current state are identified. Conflicts are resolved in favor of rules with higher strength. Rule strengths are updated based on rewards received after each training episode. The following stimuli were defined:

- *range1* .. *range9*: Value between 0.5 and 20 feet in 1-foot increments, which specifies the distance to the closest obstacle within sensors field of view.
- *range*: Value between 0 and 800 feet in 1-foot increments, which specifies the distance to the target.
- *bearing*: Value between -180 and 180 degrees in 20-degree increments, which specifies the bearing to the target.

The action parameter, turn_rate, specified the turn rate for the MAV and took on values between -20 and 20 degrees in 5-degree increments.

4.2 Learning Method

The system must learn a collision-free navigation behavior. In this study, the behaviors are evolved using the SAMUEL rule learning system. SAMUEL uses standard genetic algorithms and other competition-based heuristics to solve sequential decision problems. It features Lamarckian operators (specialization, generalization, merging, avoidance, and deletion) that modify rules based on interaction with the environment. SAMUEL has to perform a number of evaluations (on the order of 80 in the current study) in order to provide history for the Lamarckian operators, to coalesce rule strengths, and to account for the noise in the evaluations. The original system implementation and default learning parameters pertinent to evolution of rule sets are described in greater detail in [Grefenstette 1990] and [Grefenstette 1991].

4.3 Fitness Function Contribution

The fitness of the controller is proportional to the distance the MAV traveled during successful trials when the goal location is reached, or the minimum distance away from the target during an unsuccessful trial when the agent crashed or ran out of time, and contributes either [0.0-0.3] or [0.5-0.8] respectively, to the global fitness functions. The contribution is calculated as follows:

$$f_{FIT}(\vec{x}) = \begin{cases} 0.3 * \left(1.0 - \frac{D_A}{D_S}\right), & unsuccessful\,trial \\ \\ 0.5 + 0.3 * \left(\frac{D_S}{D_T}\right), & successful\,trial \end{cases}$$

where D_A is the minimum distance away from the target during the trial, D_S is an initial distance away from the target, and D_T is total distance traveled during the trial.

5 Evolution of Form

The behavior an agent adopts for a task is determined by its ability to interact and sense the environment. There are

a wide variety of sensors that could be implemented on the MAV, but the final make up of the sensor suite is constrained by the size, weight, and power capacity of the vehicle. The objective of this study is to evolve the most power-efficient sensor suite that guarantees an efficient task-specific control. Power efficiency is assumed for this study to be inversely proportional to sensing ability of the agent determined by its sensor suite coverage.

5.1 Problem Representation

Our range sensor model is based on a simple range sensor such as sonar or radar. It returns range to a single, closest obstacle in its field of view. The possible evolvable sensor characteristics include:

- range of the individual sensor;
- beam width of the individual sensor;
- placement of individual sensor on the vehicle.

In this study, the beam width and the range of each of the available sensors are being evolved. The number of sensors is evolved implicitly since values of beam width or range equal to zero imply that the sensor isn't used. Nine sensors are placed symmetrically along the direction of flight and radially from the center of the vehicle in increments of 22.5 degrees. To decrease the search space, the symmetry along the forward axis is exploited and only the forward and four sensors along one side are represented. The four sensors along the other side of the vehicle are identical to the first four. The maximum beam width of the sensor is 45 degrees while the maximum sensing range is 20.0 feet.

The sensor suite characteristics are represented as a vector of ten values: the beam width and the range for five unique sensors, each represented by a floating-point value between 0.0 and 1.0. For each sensor, the first gene value is mapped to 0 to 45 degrees that defines its beam width and the second value is mapped to 0 - 20 feet that defines its sensing range.

5.2 The Learning Method

The sensor suite characteristics are also evolved using SAMUEL. In addition to the rule set representation, SAMUEL allows a set of parameters to be attached to each of the rule sets, which we use as described above to represent the sensor characteristics. On these parameters, SAMUEL uses Gaussian mutation (mu = 0 and sigma = 0.15) and two-point crossover. It uses a fitness-proportional selection method to choose the individuals out of the population – the number of offspring is proportional to the parents fitness.

5.3 Fitness Function Contribution

The fitness of the sensor suite is inversely proportional to its coverage and contributes [0.0 .. 0.2] to the global fitness functions, but only if the agent behavior allows it to complete the task, i.e. navigate safely to the target location. The coverage of the sensor suite is calculated as the sum of the areas of the sectors defined by the beam width and the range of individual sensors. The contribution is calculated as follows:

$$f_{FORM}(\vec{x}) = 0.2 * \left(1.0 - \frac{C(x)}{C_{MAX}}\right)$$

where $C(x)$ is the coverage of the sensor suite and C_{MAX} is the maximum possible sensor coverage for the experiment; C_{MAX} is currently equal to 1413.0 square feet.

6 Experimental Design

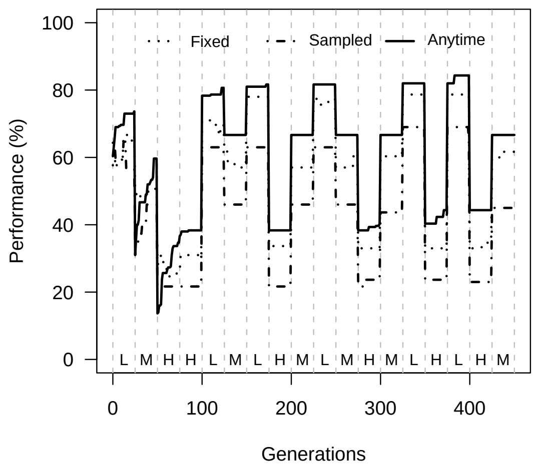
Similar to [Grefenstette 1992], we compared traditional evolution in a simulated environment with no feedback from the task environment, to case-based continuous and embedded learning in a simulated environment which reflected current state of the world. These approaches can be viewed as alternative approaches to system development; in first case, the learning is done offline in a simulation designed by the experts while in the second case, the learning is performed online after the system has been deployed. We performed three separate experiments; two baseline experiments which explored evolution in static simulation environment, and one which applied anytime coevolution of form and function technique to a dynamic simulation environment. The total length of the experiment was 450 generations with 100 members in the population. The complexity of the environment was changed every 25 generations.

6.1 Experiment 1: Fixed complexity simulation model

In this experiment, all possible solutions throughout the length of the experiment were evaluated in a series of simulated environments with the same, constant environment complexity independent of the changing environment. The tree density, which determines the complexity of the environment, was set to 2.5 trees per 100 square feet, which was previously determined to provide an adequate learning gradient and acceptable level of generalization to other densities. Whenever the learning system found a solution which outperformed the previous one in the simulation, the online strategy was updated. The changes in the environment were not registered in the simulation and the learning continued uninterrupted throughout the whole experiment.

6.2 Experiment 2: Sampled complexity simulation model

Similarly to the first baseline experiment, in this experiment, all the individuals were evaluated in a series of simulated environments of varied complexity independent of the changing environment. The tree density of the environment was chosen at random from uniform distribution of three densities, 1.25, 2.5, and 5 trees per 100 square feet. Whenever the learning system found a solution, which outperformed the previous one in the simulation, the online strategy was updated. To establish the baseline, the changes in the environment were not registered in the simulation and the learning continued uninterrupted through out the whole experiment.



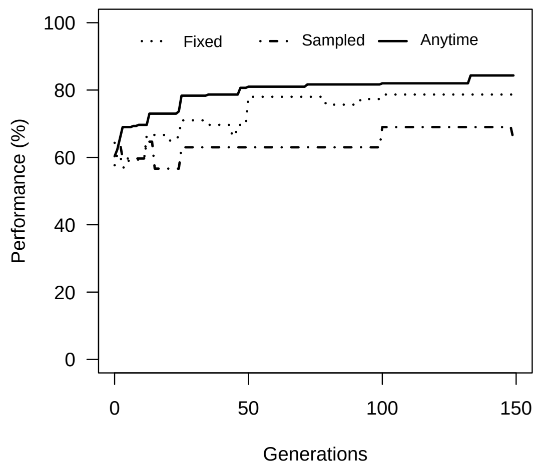


Figure 3: Summary of task performance in a changing environment.

Figure 4: Task performance in the low complexity (1.25 trees per 100 sq. ft) environment.

6.3 Experiment 3: Dynamic simulation model

In this experiment, the individuals of the current generation were evaluated in a series of simulated environments with complexity determined by the current, changing environment. The tree density of the environments varied between the same densities as in the previous model: 1.25, 2.5, and 5 trees per 100 square feet. Each density was recognized as a separate case. For the first 3 periods (25 generations each), the cases were presented in increasing order of complexity. For the rest of the experiment, the complexity of the environment within each block of three cases was selected at random. Each case was presented a total of six times. For this study, the environments were presented in the following order: L (1.25), M (2.5), H (5.0), H, L, M, L, H, M, L, M, H, M, L, H, L, H, M. Whenever the learning system found a solution which outperformed the previous one in the simulation, the online strategy was updated. When the change in the environment was detected, the simulation was updated and the offline learning was reinitialized according to case-base anytime learning strategy. On the first occurrence of the case, the population was initialized using a homogenous, simple default set of rules. The subsequent times, one half of the initial population was initialized based on a similarity metric between the current case and the previously observed cases in the case base, while the other half of the population was initialized using a default rule set. In this study, the similarity metric was simply defined as absolute difference in tree density of the environment.

7 Results

The results of anytime coevolution of form and function in a changing environment for each of the approaches described in Section 6 are summarized in Figures 3 through 6 and in Table 1.

Figures 3 through 6 show online performance of the best individuals for each approach. Each data point in the graphs represents the average performance of a best-so-far individual over 100 episodes. The data was averaged over 3 independent sets of runs for each of the baseline and the anytime learning experiments. In this study, the performance is defined as the number of times the MAV reached the goal out of a hundred. Figure 3 summarizes online performance of the system in the changing environment. The vertical lines in the plot mark the environment changes. The complexity of the environment for each period is provided along the horizontal axis; L indicates the lowest, M the medium, and H the highest tree density. Figures 4 through 6 present each level of the environment complexity individually with all the relevant periods concatenated.

The case-base continuous and embedded learning was able to outperform both alternative approaches. It is also worth noting that even though the simulation models were not updated during learning in Experiments 1 and 2, the evolved strategies were to a certain degree tolerant to the changes of the environment. Further, the strategies evolved in a simulation with a fixed complexity were more general than the ones evolved in a simulation which sampled the complexity space.

Table 1 summarizes the characteristics of the final sensor suites for each approach. The data was averaged over 3 independent sets of runs for each of the baseline and the anytime learning experiments. The beam width and the range of the five unique sensors and the total coverage of the sensor suite are presented. The goal of the evolution of form was to evolve a sensor suite with minimal coverage in order to maximize power efficiency of the vehicle which was defined to be inversely proportional to the sensor coverage.

By design, anytime learning approach allowed for higher level of specialization of sensors suites for individual cases, but it was even able to improve on the sensor suite evolved

	Sensor 1		Sensor 2		Sensor 3		Sensor 4		Sensor 5		Cov
	Width	Range	Width	Range	Width	Range	Width	Range	Width	Range	
Fixed	0.0	0.0	12.5	1.7	14.8	6.5	25.0	6.6	5.4	14.5	106.5
Sampled	15.8	10.8	7.5	3.7	3.2	5.7	29.6	11.8	2.2	10.4	108.4
Anytime (L)	23.2	9.8	14.5	10.4	16.2	12.2	30.9	6.1	5.1	13.7	108.6
Anytime (M)	17.0	10.3	16.1	8.9	27.5	4.3	11.9	4.4	3.6	11.9	82.1
Anytime (H)	14.0	7.7	18.1	8.7	12.2	6.5	9.7	1.7	2.9	14.1	70.8

Table 1: Characteristics of the sensor suites using traditional evolution in a fixed and sampled simulation environments, and using case-based anytime learning in dynamic simulation environment. Beam width and the range of the unique sensors and the total coverage of the sensor suites are presented.

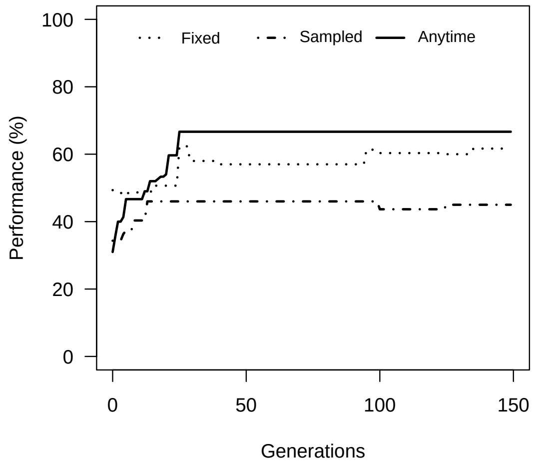


Figure 5: Task performance in the medium complexity (2.5 trees per 100 sq. ft) environment.

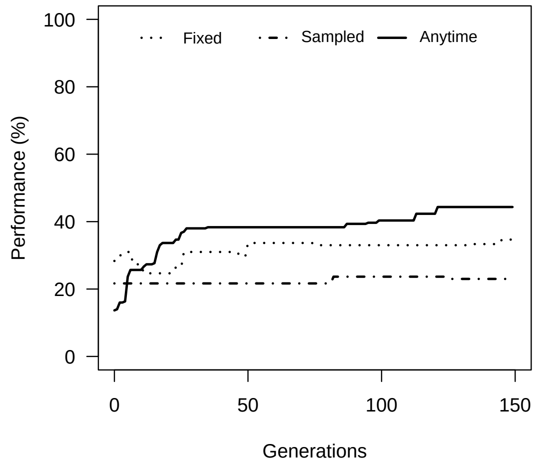


Figure 6: Task performance in the high complexity (5.0 trees per 100 sq. ft) environment.

in the static medium complexity environment. In general, the sensor suites evolved for the low density environment did not require full set of sensors and all sets included a narrow, far reaching front sensor, and several shorter side sensors. The higher density environments required more uniform distribution of sensing coverage between all available sensors.

These results show that anytime learning is a feasible approach to continuous coevolution of form and function.

8 Conclusions

In this paper, we presented an approach to continuous and embedded coevolution of form (the morphology) and function (the control behavior) for autonomous vehicles. While this study focused only on coevolution of the characteristics such as beam width and range of individual sensors in the sensor suite, and the reactive strategies for collision-free navigation for an autonomous micro air vehicle, this approach could be easily extended to evolution of more complete morphologies for more complex missions. The addition of an anytime (continuous and embedded learning) mechanism allows for more robust and adaptive systems. In particularly, it opens the door for vehicles that can morph, that is, change their configuration on the fly for different aspects of a mission or to handle unexpected situations.

Experimental results were presented which showed that continuous and embedded learning is a feasible approach to anytime coevolution of form and function. Further experiments will be performed to determine appropriate anytime learning components for the domain such as re-initialization policies or minimum case presence. We plan to extend this work to learn characteristics of an air vehicle's airframe that might be changed during a mission, such as the length of the tail structure, and the shape and geometry of the airfoils.

Acknowledgments

The work reported in this paper was supported by the Office of Naval Research under work requests N0001403WR20212 and N0001403WR20057.

Bibliography

[Balakrishnan 1996] Balakrishnan, K. and V. Honovar. On Sensor Evolution in Robotics. In Koza, Goldberg, Fogel, and Riolo (eds.), Proceedings of 1996 Genetic Programming Conference GP-96; MIT Press, pp. 455-460; 1996.

[Bugajska 2000] Bugajska, Magdalena D. and Alan C. Schultz. (2000) ”Co-Evolution of Form and Function in the Design of Autonomous Agents: Micro Air Vehicle Project.” In Proceedings of GECCO-2000 Workshop on Evolution of Sensors in Nature, Hardware, and Simulation; Las Vegas, NV, July 8, 2000.

[Bugajska 2002] Bugajska, Magdalena D. and Alan C. Schultz. (2002) ”Coevolution of Form and Function in the Design of Micro Air Vehicles.” In Proceedings of 2002 NASA/DoD Conference on Evolvable Hardware (EH-2002); Washington, DC, July 15-18, 2002.

[Cliff 1993] Cliff, D., I. Harvey, and P. Husbands ”Explorations in Evolutionary Robotics”, Adaptive Behaviour (MIT Press) 2(1):71–108, 1993.

[Cliff 1996] D. Cliff and G. F. Miller ”Co-evolution of Pursuit and Evasion II: Simulation Methods and Results” in: P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. Wilson (editors) From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96). MIT Press Bradford Books, pp.506–515, 1996.

[Floreano 1996] Floreano, D. and F. Mondada. Evolution of Homing Navigation in a Real Mobile Robot. In IEEE Transactions on System, Man, and Cybernetics Part B; 26(3) 396-407; 1996.

[Funes 1997] Funes, P. and J. Pollack. Computer Evolution of Buildable Objects. In Husbands and Harvey (eds.), Proceedings of The Fourth European Conference on Artificial Life, MIT Press, pp.358-367; 1997.

[Grefenstette 1990] Grefenstette, J. J., Ramsey, C. L. and Schultz, A. C. (1990). Learning sequential decision rules using simulation models and competition. Machine Learning 5(4), 355-381.

[Grefenstette 1991] Grefenstette, J. J. The Users Guide to SAMUEL, Version 1.3. NRL Memorandum Report 6820, Naval Research Laboratory, 1991.

[Grefenstette 1992] Grefenstette, J. J. and C. L. Ramsey. (1992) ”An approach to anytime learning.” In Proceedings of Ninth International Conference on Machine Learning, pp. 189-195, Morgan Kaufmann, San Mateo, CA.

[Harvey 1992] Harvey, I., P. Husbands, and D. Cliff. Issues in Evolutionary Robotics. Proceedings of the Second International Conference on Simulation of Adaptive Behaviour; MIT Press Bradford Books, 1993.

[Hornby 2001] Hornby, G. S. and J. B. Pollack (2001). Body-Brain Co-evolution Using L-systems as a Generative Encoding. In Proceedings of the Genetic and Evolutionary Computation Conference 2001, San Francisco, CA; pp. 868-875.

[Husbands 1996] Husbands, P., G. Jermy, M. McIlhagga, and R. Ives. Two Applications of Genetic Algorithms to Component Design. In selected papers from AISB Workshop on Evolutionary Computing; Fogarty, T (ed.), Springer-Verlog; Lecture Notes in Computer Science, 1996.

[Lee 1996] Lee, Wei-Po, J. Hallam, and H. H. Lund. A Hybrid GP/GA Approach for Co-evolving Controllers and Robot Bodies to Achieve Fitness-Specific Tasks. In Proceedings of IEEE Third International Conference on Evolutionary Computation, IEEE Press, NJ,;1996.

[Lichtensteiger 1999] Lichtensteiger, L. and P. Eggenberger. Evolving the Morphology of a Compound Eye on a Robot. Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot 99); 1999.

[Lund 1997] Lund, H. H., J. Hallam, and W-P. Lee. Evolving Robot Morphology. Proceedings of IEEE Fourth International Conference on Evolutionary computation; IEEE Press, NJ, 1997.

[Lund 1998] Lund, H. H. and O. Miglino. Evolving and Breeding Robots. In Proceedings of First European Workshop on Evolutionary Robotics; Springer-Verlag, 1998.

[Mark 1998] Mark, A., D. Polani, and Thomas Uthmann. A Framework for Sensor Evolution in a Population of Braitenberg Vehicle-like Agents. In C. Adam, R. Belew, H. Kitno, and C. Taylor (eds.), Proceedings of Artificial Life IV; pp. 428-432; 1998.

[Nolfi 1994] Nolfi, S., Floreano, D., Mighno, O., and F. Mondada. How to Evolve Autonomous Robots: Different Approach in Evolutionary Robotics. In R. Brooks and P. Maes (eds.), Proceedings of the International Conference Artificial Live IV; MIT Press, pp.190-197; 1994.

[Potter 2001] Potter, Mitchell A., Lisa A. Meeden, and Alan C. Schultz. Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists. In Proceedings of The Seventeenth International Conference on Artificial Intelligence, Seattle, WA; August 4-10, 2001.

[Ramsey 1994] Ramsey, C. L. and J. J. Grefenstette. (1994) ”Case-based anytime learning.” In Case Based Reasoning: Papers from the 1994 Workshop, D. W. Aha, ed., Technical Report WS-94-07, AAAI Press, Menlo Park, CA.

[Schultz 1996] Schultz, A. C., J. J. Grefenstette, and William Adams. RoboShepherd: Learning a complex behavior. Presented at RobotLearn96: The Robotics and Learning workshop at FLAIRS 96; 1996.

[Schultz 2000] Schultz, Alan C. and J. J. Grefenstette. (2000) ”Continuous and Embedded Learning in Autonomous Vehicles: Adapting to Sensor Failures.” In Proceedings of Unmanned Ground Vehicle Technology II, (Eds. Grant R. Gerhart, Robert W. Gunderson, Chuck M. Shoemaker), Proceedings of SPIE Vol.4024, pg. 55-62, 2000.

[Sims 1994] Sims, K. Evolving 3D Morphology and Behavior by Competition. In R. Brooks and P. Maes (eds.), Proceedings of the International Conference Artificial Live IV; MIT Press, Cambridge, MA, pp.28-39; 1994.